PROGETTO S11/L5

TRACCIA

Con riferimento al codice presente nelle slide successive, rispondere ai seguenti quesiti:

1. Spiegate, motivando, quale salto condizionale effettua il Malware.
2. Disegnare un diagramma di flusso (prendete come esempio la visualizzazione grafica di IDA) identificando i salti condizionali (sia quelli effettuati che quelli non effettuati). Indicate con una linea verde i salti effettuati, mentre con una linea rossa i salti non effettuati.
3. Quali sono le diverse funzionalità implementate all'interno del Malware?
4. Con riferimento alle istruzioni «call» presenti in tabella 2 e 3, dettagliare come sono passati gli argomenti alle successive chiamate di funzione . Aggiungere eventuali dettagli tecnici/teorici

PUNTO 1

SPIEGAZIONE SALTO CONDIZIONALE #1

Il valore 5 viene trasferito nel registro EAX utilizzando il comando “mov”.
Successivamente, con il comando “cmp”, il contenuto di EAX viene confrontato con il valore 5.

Dato che i due valori coincidono, il salto a un'altra locazione non avviene (il comando “jnz” significa “salta se non zero”, ma poiché il confronto restituisce 1, il salto a “0040BBA0” non viene eseguito).

Ecco cosa accadrebbe se il salto venisse effettuato:

Alla locazione “0040BBA0", il contenuto di “EDI” (che contiene un link malevolo) verrebbe copiato in “EAX”, sostituendo il valore 5 inizialmente presente.
Successivamente, “EAX” verrebbe inserito nello stack tramite il comando “Push” (dove “EAX” rappresenta il file .exe da eseguire).

Infine, verrebbe eseguita la funzione “DownloadToFile()” tramite il comando “call”, che scaricherebbe un file .exe malevolo dal sito indicato dal link.

PUNTO 1

SPIEGAZIONE SALTO CONDIZIONALE #2

Inizialmente, il valore 10 viene assegnato al registro “EBX” ma successivamente, questo valore viene incrementato di 1 (10+1) utilizzando il comando “inc”.

A questo punto, con il comando “cmp”, il valore di “EBX” viene confrontato con 11, e poiché il comando “jz” significa “salta se zero”, il salto alla locazione “0040FFA0” avviene perché il confronto restituisce 1.

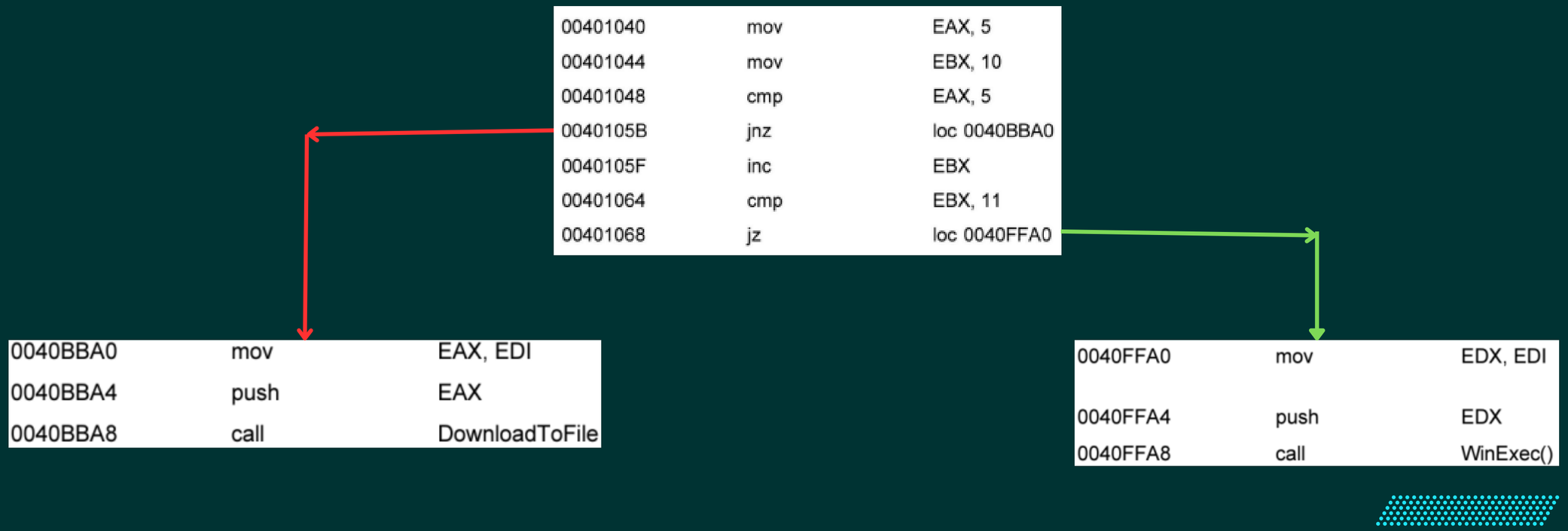
Nella locazione “0040FFA0”, il contenuto di EDI viene copiato nel registro “EBX” tramite il comando “mov” (dove EDI contiene il percorso di un file .exe malevolo).

Il registro “EBX” viene poi inserito nello stack con il comando “push”, e infine, il file viene eseguito (in questo caso, si tratta probabilmente di un ransomware) tramite il comando “call” che richiama la funzione “WinExec()” per eseguire il ransomware.

PUNTO 2

DIAGRAMMA DI FLUSSO

Di seguito potete trovare il diagaramma di flusso che mostra sia il salto effettuato, evidenziato dalla linea verde, sia il salto non effettuato evidenziato dalla linea rossa.



PUNTO 3
FUNZIONALITÀ ALL'INTERNO DEL MALWARE

La funzione WinExec è utilizzata per avviare un file .exe creando un nuovo processo che si occuperà della sua esecuzione.

Il primo parametro è una stringa che indica il percorso del file da eseguire, mentre il secondo parametro specifica come deve essere visualizzata la finestra dell'applicazione (ad esempio, nascosta, minimizzata o massimizzata).

Internamente, la funzione si occupa di caricare ed eseguire l'applicazione, gestendo l'ambiente di esecuzione del nuovo processo creato. Il valore di ritorno della funzione indica se l'esecuzione è avvenuta con successo o meno.

PUNTO 3
FUNZIONALITÀ ALL'INTERNO DEL MALWARE

La funzione DownloadToFile recupera dati (file di qualsiasi tipo) da un sito web e li salva in un file sul sistema in cui la funzione viene eseguita.

Essa scarica il file dalla posizione specificata, richiedendo sia l'indirizzo della risorsa remota che il percorso locale in cui salvare il file.

Durante il processo, dopo essere stata chiamata, la funzione si connette alla risorsa remota, acquisisce i dati e li scrive nel file di destinazione.

PUNTO 4
ARGOMENTI FUNZIONI “CALL”

In entrambi i casi di salto, i parametri per l'istruzione “call” vengono trasferiti tramite lo stack.

Questo processo inizia con l'uso dei comandi “mov”, “push” e infine “call”, che vengono utilizzati in sequenza per i salti condizionali. Ecco cosa rappresenta ciascuno di questi comandi:

- mov: copia un dato o una variabile all'interno di un registro.
- push: inserisce una variabile o un registro nello stack.
- call: richiama una funzione specifica, che può essere di diversa natura (come WinExec, DownloadToFile, ecc.).

DETTAGLI TECNICI/TEORICI

“PUSH”

Push di dati:

Prima di chiamare una funzione con l'istruzione call, è comune che vengano pushati i parametri necessari nello stack. Questi parametri possono includere variabili o altri dati che la funzione richiede per svolgere il suo compito.

“CALL”

Chiamata di funzione:

Dopo che i dati sono stati pushati nello stack, l'istruzione call viene utilizzata per chiamare la funzione desiderata.

“ESECUZIONE DELLA FUNZIONE”

Dopo che è stata effettuata la chiamata alla funzione, il controllo passa alla funzione stessa. La funzione esegue le operazioni necessarie, accedendo ai dati pushati nello stack se necessario

THANKYOU